

PERFORMANCE EVALUATION OF TREE-BASED AND MULTIPATH-BASED MULTICAST ROUTING FOR NETWORKS-ON-CHIP

Faizal Arya Samman

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Hasanuddin
Jl. Perintis Kemerdekaan Km. 10, Makassar 90245, Indonesia.

E-Mail: faizalas@unhas.ac.id

Abstract—Multicast routing is one of collective communication supports for parallel computing systems. This paper discuss the issue in network or low-level layers. The multicast routing can be implemented using tree-based and path-based methods. This paper presents performance evaluations of deadlock-free tree-based and multipath-based multicast routing for networks-on-chip (NoC). Multicast packets are switched and scheduled in the NoC using a local identity tag-based multiplexing technique. The local ID-tag attached to every flit is updated over communication links allowing different flits of different packets to be mixed in the same queue. By using a fair flit-by-flit round arbitration and multicast contention management to share communication channels, deadlock in intermediate nodes as a main problem in the tree-based and multipath-based multicast routing can be handled efficiently and effectively.

I. INTRODUCTION

Multicast delivery services have been intensively used in large-scale multiprocessor systems, and have been fundamental services of some data parallel computer languages [1]. The use of multicast services in numerous parallel algorithms, e.g. parallel search and parallel graph algorithms, has been shown to benefit. In a single-program multiple-data (SPMD) mode of computation, multicast communication is of benefit. The same program is executed on different processors with different data, and several data are proceeded in parallel. In a data parallel mode of computation, a variety of process control operations and global data movement such as reduction, replication, permutation segmented scan and barrier synchronization requires collective communication models. In a distributed shared-memory paradigm, multicast services may be used to efficiently support shared-data invalidation and updating. The multicast delivery can be implemented with software approach by sending a separate copy of the messages from the source to every destination node (unicast-based multicast delivery). However, this approach is unefficient in terms of communication latency and energy.

Our NoCs, which is called XHINoC (eXtendable Hierarchical and Irregular NoC) has been synthesized at gate-level and supports both tree-based and multi path-based multicast routing services. XHINoC switches packet in the network by using a novel switching method called “wormhole cut-through (WCT) switching method” [16]. The WCT switching method can solve effectively head-of-line problems that usually occur when we use a traditional wormhole switching method.

There are some network-on-chip prototypes that have been implemented so far. Æthereal NoC [2] and Nostrum

[3] for example have used a time-division-multiplexing method in order to be able to support further the multicast services. However, experiments by analysing multicast traffics and the NoCs performances over multicast deadlocks have not been released so far.

II. RELATED WORKS AND MOTIVATIONS

Multicast messages can be routed in the network using path-based [1], [4], [5] or tree-based [6], [7], [8] multicast routing. A higher probability that a multicast deadlock occurs in intermediate nodes has alleviated the intentions of the most of network designers to use tree-based multicast routing. However, the deadlock configuration can also occur theoretically in destination nodes when using path-based multicast routing, and particularly also in intermediate nodes when using dual-path or multipath routing. Therefore, a new multicast contention management and scheduling policy for effective deadlock-free multicast routing is presented in this paper. Moreover, the multicast networks presented in the abovementioned works are not dedicated for single-chip networks, and the routing hardware units presented in those works are very complex, and may also increase the logic area after gate-level synthesis. In our NoCs, the adaptive routing algorithms used to route unicast and multicast packets are the same resulting in a very efficient routing function gate-level implementation.

The NoC presented in [9] has introduced the path-based multicast routing to avoid multicast deadlock in the destination nodes by reserving virtual channels and giving priority for the multicast message over the unicast message on arbitration of link bandwidth. Experiments in the work show that the proposed multicast technique improves throughput, and does not exhibit significant impact on unicast performance in a network with mixed unicast-multicast traffic “only if” the network is not saturated. Our proposed multicast scheduling does not give priority for multicast messages (fair flit-by-flit arbitration between the unicast and multicast messages). Hence, the significant impact on the unicast performance is not presented “even if” the network is saturated. Indeed, the NoC in [9] has not been synthesized into logic gate level.

The NoC presented in [10] uses a time-space-time switch designed for time-division-multiplexing (TDM-based) NoCs. Slot map tables as central components are used as time slot interchangers to directly control the read and write operation to random access frame buffers. Unfortunately, although this work has mentioned the feasibility of implementing the multicast scheduling technique, a concrete multicasting procedure, system-level or RTL-level simulations for measuring the NoC

performance over multimessage multicast traffics and the NoC capability to handle the multicast deadlock are not presented in the paper.

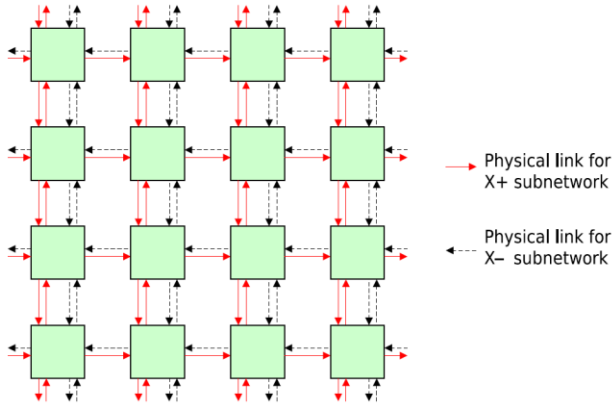


Figure 1. 2-D Mesh Planar topology.

III. MULTIPATH AND TREE-BASED MULTICASTING

A. 2D Mesh Planar Topology

Fig. 1 presents an example of the 2-D mesh planar 4x4 network. The network is physically divided into two subnetworks i.e., X+ (depicted in solid arrows) and X- subnetworks (depicted in dashed arrows). If the x-distance between source and target nodes ($X_{\text{offset}} = X_{\text{target}} - X_{\text{source}}$) is zero or positive, then packets will be routed through the physical channels of the X+ subnetwork. If X_{offset} is zero or negative, then the packets will be routed through the physical channels of the X- subnetwork. We can assume that the ports connected with vertical links of X+ and X- subnetworks are denoted by (North1, South1) and (North2, South2) ports, respectively. Hence, the packets routed through the X+ subnetwork will have adaptivity to make West-North1, West-South1, North1-East and South1-East turns as well as West-East, North1-South1 and South1-North1 non-turn routing. While the packets routed through the X- subnetwork will have adaptivity to make East-North2, East-South2, North2-West and South2-West turns as well as East-West, North2-South2 and South2-North2 non-turn routing directions.

The planar adaptive routing on a mesh topology is firstly introduced in [11] and deadlock-free. Instead of using virtual channels to implement the link interconnect between NORTH and SOUTH port as made in [11], we prefer to implement two physical channels to separate the NORTH-SOUTH link interconnects for X+ and X- subnetworks. The objectives of this approach are to maintain the router performance and to increase the network bandwidth. If the virtual channels are implemented in the NORTH and SOUTH ports, then we need to add two virtual queues at both incoming and outgoing ports.

Rather than using such virtual queues, which can degrade router performance or increase data transfer latency, we substitute them by adding additional ports (NORTH2 and SOUTH2 ports) in the existing mesh router as presented in Fig. 3(b). In this approach, the number of additional queues is similar to the virtual channel implementation but it maintains the router

performance. Nevertheless, the number of input-output pins is certainly increased. More explanations on the planar adaptive routing algorithm under planar network topology can be found in [17] and [18].

B. Multicast Packet Format

Fig. 2(a) shows a multicast packet format for the multipath-based multicast routing. In this multicasting method, the destination nodes are grouped into four areas relative to an injection node X, Y i.e., Quadrant Area 1 ($X_{\text{offset}} > 0$ and $Y_{\text{offset}} \geq 0$), Area 2 ($X_{\text{offset}} \leq 0$ and $Y_{\text{offset}} > 0$), Area 3 ($X_{\text{offset}} < 0$ and $Y_{\text{offset}} \leq 0$) and Area 4 ($X_{\text{offset}} \geq 0$ and $Y_{\text{offset}} < 0$). The first packet header representing the closest node in the multicast group is assigned as a Header (HEAD) flit. The other packet headers for other destination nodes located in the same group are assigned as Header-Tail (HTail) flits. The use of such flit types is aimed at simplifying the multipath-based multicast routing that will be explained in Subsection 3.C.

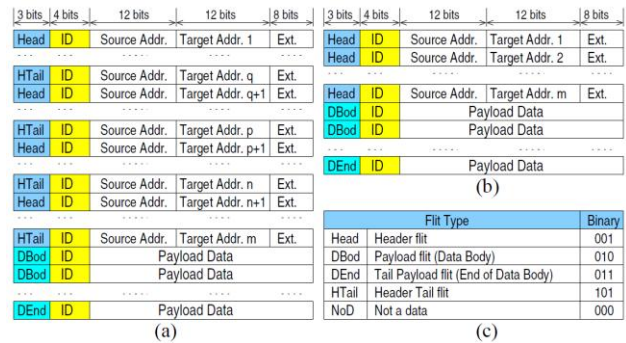


Figure 2. Multicast packet format.

The Packet format for tree-based multicast routing is simple as depicted in Fig. 2(b). Each flit containing target nodes of a multicast node is assigned as a header flit, while the payload flits including the last payload flit are assigned as DBody and DEnd flits. A message in our NoC is associated as single packet with a header flit or header groups for multicast message sent into the NoC before payload flits. This format choice can guarantee in-order delivery when runtime adaptive routing algorithms are utilized. Fig. 2(c) presents the binary coding for the types of the flits, i.e. header (Head), header tail (HTail), databody (DBod) and tail (DEnd) flits.

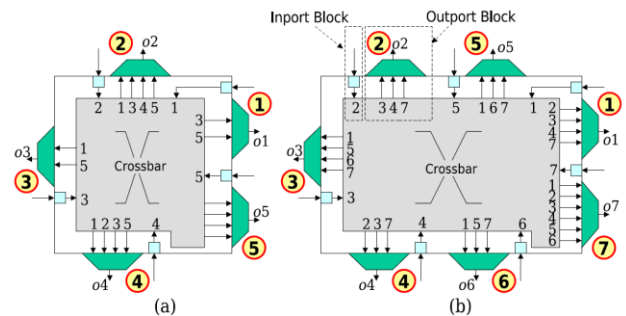


Figure 3. Switch Structure. (a) Mesh Static XY routing, (b) Mesh Planar Adaptive Routing.

C. Multicast Routing Algorithms

In this paper, three NoC router prototypes will be designed with different routing schemes to evaluate the area-performance trade-off of the multicast routers i.e., static XY tree-based multicast routing using a standard mesh router as shown in Fig. 3(a), as well as planar adaptive tree-based and multipath-based multicast routing using mesh planar router as shown in Fig. 3(b). In the static XY tree-based routing (TB XY), packets are routed firstly to horizontal X-direction and then to vertical Y-direction. Hence, North-East, North-West, South-East and South-West turns are prohibited. In the planar adaptive tree-based routing (TB PA), the number of available free ID slots and routing directions that have been made by the same previous header flit are used as the second and the first priority weighting signals to decide for an adaptive routing direction. Packets from the LOCAL port that can be routed adaptively to horizontal or vertical direction will be prioritized to select the vertical direction (i.e. a situation where the numbers of free ID slots are similar in both outports and routing direction for similar packet has not been made previously to horizontal direction). While from any port (except from LOCAL port) the packets prefer to make non-turn routing direction if the packets can be routed adaptively to horizontal or vertical direction.

In the multipath-based multicast routing (MPB), four routing paths are performed in the four multicast groups as mentioned in Subsection 3.2. In each group, the first packet header (HEAD flit type) is routed to its destination node followed by other header tails (HTail flit type), payload data (DBody flit type) including the payload tail (DEnd flit type) flits. When the first header (HEAD flit) reaches its destination node from a certain inport, the LOCAL routing information is assigned in a routing table and forwarded to a routing hardware unit in the inport. Then, when the first header tail (HTail flit) comes into the inport, the routing hardware will find appropriate routing direction for this header tail (it will not follow again the header flit that has been routed to the LOCAL port), and then its Type-field bits (HTail) is updated to HEAD flit type. Hence, the remaining flits behind will follow routing paths performed by this header flit until the header reaches also its destination node.

D. Multicast Destination Ordering

In the static XY tree-based multicast routing, the header ordering in source nodes is not required (the order of the destination addresses can be freely determined). The multicast routing will form communication paths like branches of trees connecting the source node with the destination nodes at the end points of the tree branches. In the planar adaptive tree-based multicasting, the destination ordering is basically not a must. However, based on our experience a good header ordering can optimize the number of tree-branches performed adaptively by the router. There are many schemes to order the packet headers. In this paper, we use a simple dual-area dimension-order shortest distances i.e., the targets nodes are firstly grouped into X- and X+ targets. The destinations are firstly sorted based on the smallest X_{offset} relative to the X_{source} address (destination nodes located in X_{offset} -distance closer to the X_{source} will be sorted earlier). Afterwards, for each destination in the X_{offset} distance subgroups, they are sorted again based on the smallest

Y_{offset} relative to the Y_{source} address. Nevertheless, in order to avoid a double tree-branch in the vertical links related to the source node, the target nodes which have X_{target} address similar to X_{source} are sorted as the last destination nodes.

In the multipath-based multicasting, the packet headers are grouped into four groups as explained in Subsection 3.B. The destinations in each group are then sorted based on the smallest X_{offset} relative to the X_{source} address. However, the order of packet headers in each X_{offset} -distance subgroup can be sorted increasingly or decreasingly depending on the location of the closest destination node in the next X_{offset} -distance subgroup from the latest destination in the current X_{offset} distance subgroup.

E. On-Chip Router Architecture

Fig. 3 shows the switch structures for networks with standard mesh and mesh planar topology. In the mesh standard, the EAST, NORTH, WEST, SOUTH and LOCAL ports are represented by port numbers 1, 2, 3, 4 and 5, respectively. While in the mesh planar, the EAST, NORTH 1, WEST, SOUTH 1, NORTH 2, and SOUTH 2 and LOCAL ports are represented by port numbers 1, 2, 3, 4, 5, 6 and 7, respectively. The numerical numbers in the crossbar area represents the connectivity between links from the incoming ports to the outgoing multiplexors.

Our NoC router microarchitecture is developed based on modular units and is grouped into incoming block and outgoing block components. Each module contains generic codes, which are strongly related to the number of input-output connectivities of each port. Fig. 4 shows the incoming and outgoing components in the Port 2 (NORTH 1 port) of the mesh planar router for instance. In the incoming block, we need 3-input GMC (Grant-Multicasting Controller) and RDec (Request Decoder), because the data coming from Port 2 is only connected to outgoing Port 1, 4 and 7. The GMC itself is an important unit to control the multicast flit release from the FIFO buffer. In the outgoing block, we need 3-input Arb (Arbiter), RFC (Request-Feedback Controller), WDec (Winner-out-Decoder) and 3-input outgoing multiplexor, because the data going out to Port 2 are from incoming Port 3, 4 and 7. The RFC unit is used to control multicast requests for avoiding improper multicast flit replications.

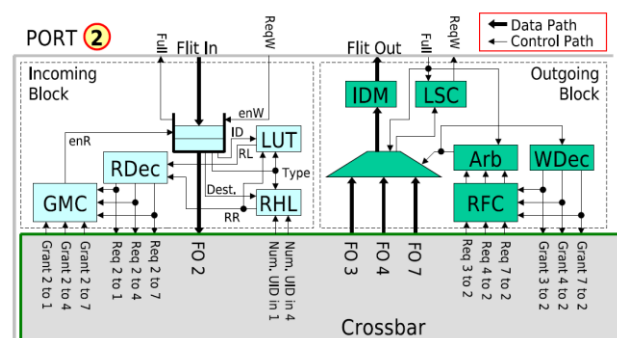


Figure 4. Architecture and components in Port 2 (NORTH 1).

The design of our NoC routers is fully customized on demand. However, each VHDL entity contains generic codes, which enable us to derive a new VHDL module

with different behavioral architecture and the number of input/output pins according to the specification. The custom-genericmodular-based design approach enables us to develop easily irregular NoC topologies.

There are different structures and intermodule signaling of the planar adaptive tree-based and multipath-based routing between the RHL (Routing hardware logic), LUT (Routing look-up table) and the RDec modules, which are not presented in this paper for the sake of simplicity. The different implementation is affected by the routing behaviors of both multicast routings that have been explained in Section 3.C.

IV. ID-BASED SWITCHING ORGANIZATION

A. ID Slot Allocations

In our NoC, unicast and multicast messages are multiplexed at each outgoing link based on an ID slots allocation technique. As a counterpart of a Time-Division Multiplexing (TDM) technique, our ID-based Multiplexing technique provides more flexible and optimistic solution for scheduling unicast or multicast message in networks at runtime. There is also no need for a global network view if the link would be scheduled preferably at design time.

Fig. 5 shows how ID-slots of each outgoing link are allocated for each tree-branch of the multicast packets A, B and C which are injected into the mesh 4x4 NoC topology. As shown in the figure, each tree-branch of the multicast message has different local ID-tag. The local ID-tags are updated over the links by using an ID mapping management technique as later explained in Subsections 4.B. More details on the ID-based routing organization can be found in [13], [12]. Some multicast messages have also contentions to access the same outgoing links in the figure, in which multicast deadlocks are performed. In order to overcome that problem, we introduce a fair flit-by-flit hold-release scheduling policy as presented later in Subsection 4.C.

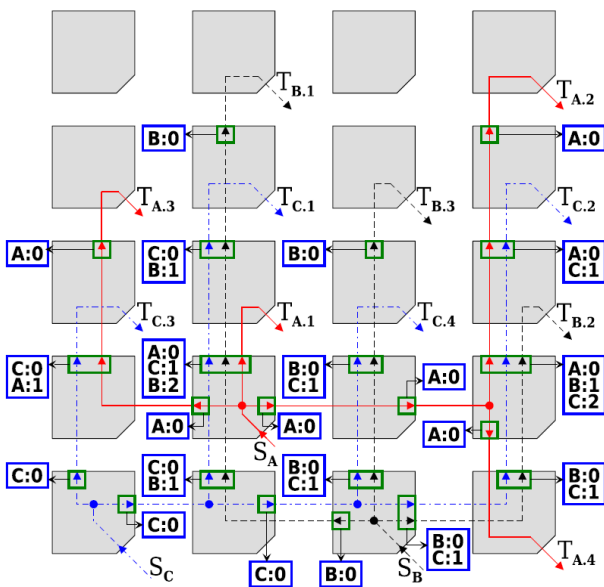


Figure 5. ID-tag slot allocation of tree-based multicast packet routing.

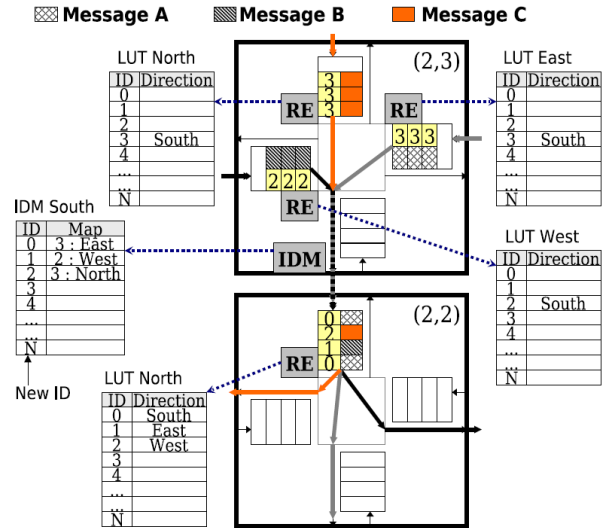


Figure 6. ID-based routing and packet mixing.

B. ID-based Routing Organization

The ID-based scheduling technique enables us to mix different flits of different messages in the same queue and to perform a fair flit-by-flit round arbitration to share outgoing ports. Fig. 6 presents three messages (Messages A, B and C coming from EAST, WEST and NORTH port, respectively) in the router node (2,3) that are switched to the router node (2,2) through the same SOUTH outgoing port. For the sake of simplicity, only routing tables (LUT) of the routing engine (RE) units of the occupied incoming ports are presented in the figure. Message A, B and C have local ID-tag 3, 2 and 3, respectively. Hence, the SOUTH routing direction are indexed and addressed in the routing tables based on the ID-number.

An ID management (IDM) unit at the SOUTH outgoing link as shown in Fig. 6 is used to update the local ID-tag of each packet into a new ID-tag before entering the next downstream router. Each new packet is allocated into a free ID slot and indexed/mapped based on its old local ID-tag and from which port it comes. As presented in the figure, Message A, B and C are mapped from local ID-tags 3, 2 and 3 into new local ID-tags 0, 1 and 2, respectively.

C. ID-Based Multicast Contention Management

Deadlock configurations because of multicast contentions can occur not only in tree-based but also in multipath-based multicast routing. The deadlock occurs in an intermediate node when one or more outgoing links are simultaneously requested by the same multicast packet. Therefore, we propose a novel scheduling method to manage the multicast contention. More detail on how the multicast contention problem can be solved by XHiNoC is presented very clear in [12] and [14]. A multicast scheduling method and a fair flit-by-flit round arbitration of a so called hold-release multicast fair scheduling policy for the deadlock handling mechanism are depicted conceptually in [14] and formally in [15].

V. SIMULATION RESULTS

In this section two simulation scenarios are presented, i.e. a simple traffic scenario with single multicast source and a complex random scenario with multiple multicast sources. Three on-chip networks with three different router prototypes are compared, i.e. a tree-based multicast router with static XY routing algorithm (TB XY), a tree-based multicast router with planar adaptive routing algorithm (TB PA) and a multipath-based multicast router (MPB).

A. Single Source Scenario

Fig. 7 shows a simple scenario to evaluate the performance and communication energy of the three multicast router prototypes. Because only one multicast message injected to node (3,4) and ejected from 18 destination nodes, then it is simple for us to analyse the routing paths and trees (traffics), the number of occupied nodes and links as well as the maximum hops to transmit data to the destination nodes. 2048 flits are injected to the source node S1. Hence, 36558 flits are accepted in the $m = 18$ destination nodes (denoted with symbol T1.m). The number cycle periods to accept the last flit at each destination node is presented in Fig. 8.

Communication energy, E_{Com} , can be accumulated from the energy to store data in network router (node), E_R , and in link (internode channel), E_L ($E_{Com} = E_R + E_L$). As presented in Fig. 7, the MPB multicasting requires more communication

energy than the other two routing techniques (it uses 43 nodes, including source node, and 42 links (channels) to transfer the multicast message). While the TB PA multicasting consumes the least communication energy (it uses 35 nodes and 34 links).

The communication energy is strongly dependent on the pattern of the destination nodes and the selected routing decisions. As shown in Fig. 7 for instance, the TB XY routing consumes less transfer energy than the MPB routing for the destination pattern performed by T1.1, T1.2, T1.3, T1.4 and T1.5. But less transfer energy is given by the MPB routing compared with the TB XY routing for the destination pattern performed by T1.10, T1.11 and T1.12. The TB PA itself takes the advantage of its routing adaptivity by making an optimal routing decisions in both destination patterns.

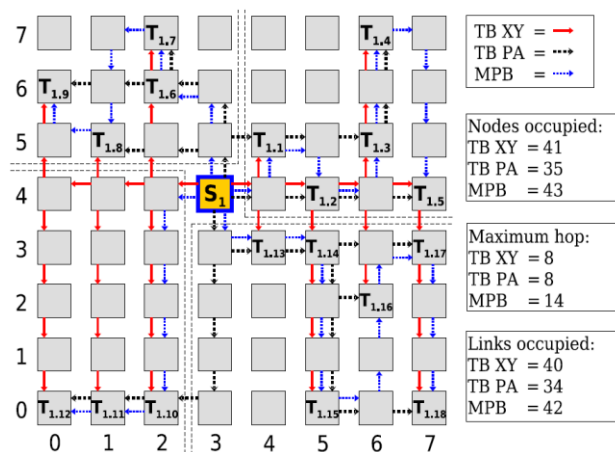


Figure 7. Scenario with one multicast source.

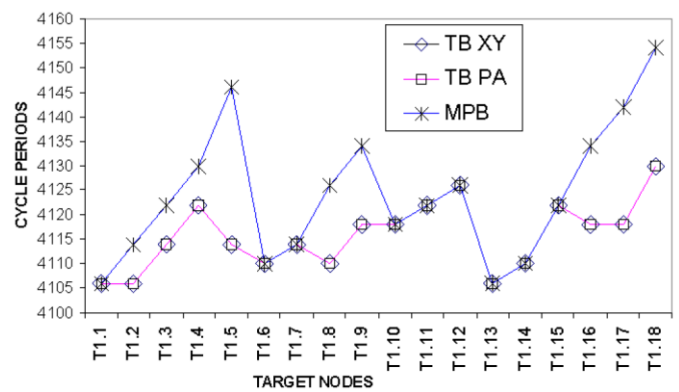


Figure 8. Simulation results of scenario in Fig. 7.

B. Complex Random Scenario

A more complex scenario used to evaluate the threemulticast routings is presented in Fig. 9. Multicast and unicast messages are injected from 10 (S1 until S10) and 5 (S11 until S15) source nodes, respectively. Each multicast message injected from S_n node will be then accepted from $m=6$ destination nodes (denoted by symbol Tn.m). Four multicast source nodes (S1, S2, S3 and S4) is selected in the center of the NoC, and their target nodes are selected randomly in the related vertical and horizontal addresses of the source nodes. In this configuration, multicast deadlock will be performed not only when using tree-based routing but also when multipath-based routing is used. The remaining source and target nodes of the multicast and unicast messages are selected randomly in the network. Although the traffic pattern does not represent an example of a real application, we are sure that the scenario can be accepted as one of the best-case scenario to verify our methodology.

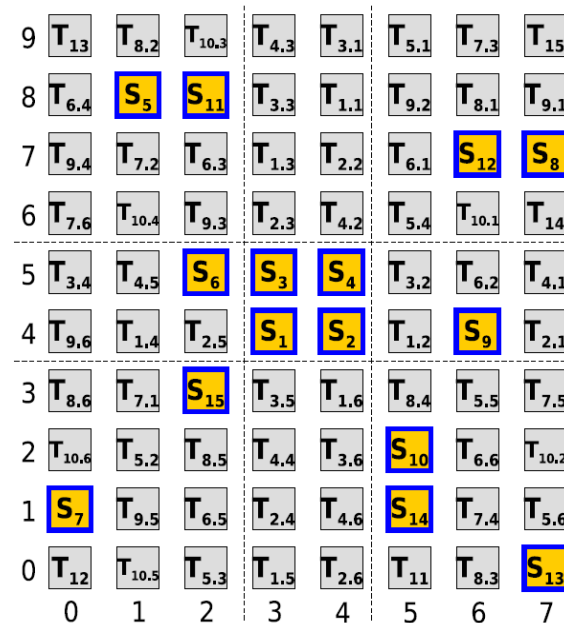


Figure 9. Complex random traffic scenario.

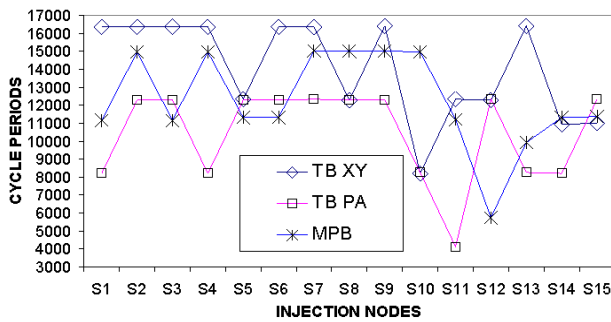


Figure 10. Simulation results of scenario in Fig. 9.

This traffic pattern performs some multi deadlock configurations because of contentions of some multicast messages to acquire the same links. 2048 flits are injected to each source node. Hence, 132820 flits $((6 \times 10 \times 2043) + (5 \times 2048))$ are accepted in the 60 (10×6) destination nodes. Each flit belonging to the same message injected from a certain node is encoded to recognize and differentiate it from other multicast messages. Every flit is then numbered in-order to enables us to check the flits one-by-one in our testbench program, whether any flit looses or is replicated improperly in the network or is accepted out-of-order in the destination nodes.

Our experiments have successfully proved our methodology, in which all flits are accepted looseless in the destination nodes without out-of-order problem and no improper flits replication. We have measured the number of cycle periods to accept the last flit at each destination node. The tail flits transfer latency (in clock cycle) of the unicast messages and the maximum latency values selected among 6 tail flits accepted from the 6 target nodes as data representations for each multicast message injected from source nodes S_n are presented in Fig. 10. It looks that the NoC with the tree-based planar adaptive multicast router (TB PA) shows the best performance over the tree-based static XY (TB XY) and multipath-based (MPB) multicast router prototypes. While the MPB prototype gives smaller tail flit transfer latency than the TB XY prototype, because the number of contentions by using MPB router can be minimized.

REFERENCES

- [1] X. Lin, P. K. McKinley and L. M. Ni, "Deadlock-Free Multicast Wormhole Routing in 2-D Mesh Multicomputers," IEEE Trans. On Parallel and Distributed Systems, vol. 5, no. 8, pp. 793–804, Aug. 1994.
- [2] E. Rijpkema, K. Goossens, A. Radulescu, J. Dielissen, J. van Meerbergen, P. Wielage and E. Waterlander, "Trade-offs in the design of a router with both guaranteed and best-effort services for networks on chip," IEE Proc. Computers and Digital Techniques, vol. 150, no. 5, pp. 294–302, Sep. 2003.
- [3] M. Millberg, E. Nilsson, R. Thid and A. Jantsch, "Guaranteed Bandwidth using Looped Containers in Temporally Disjoint Networks within the Nostrum Network on Chip," Proc. Design, Automation and Test in Europe Conf. and Exhibition (DATE'04), pp. 890–895, 2004.
- [4] J. Duato, "A Theory of Deadlock-Free Adaptive Multicast Routing in Wormhole Networks," IEEE Trans. on Parallel and Distributed
- [5] R. V. Boppana, S. Chalasani and C. S. Raghavendra, "Resource Deadlocks and Performance of Wormhole Multicast Routing

- Algorithms," IEEE Trans. on Parallel and Distributed Systems, vol. 9, no. 6, pp. 535–549, June 1998.
- [6] M. Barnett, D. G. Payne, R. A. van de Geijn and J. Watts, "Broadcasting on Meshes with Worm-Hole Routing," Journal of Parallel and Distributed Computing, vol. 35, no. 2, pp. 111–122, 1996.
- [7] M. P. Malumbres, J. Duato and J. Torrelas, "An Efficient Implementation of Tree-Based Multicast Routing for Distributed Shared-Memory Multiprocessors," in Proc. of the 8th IEEE Symposium on Parallel and Distributed Processing, pp. 186–189, 1996.
- [8] D. R. Kumar, W. A. Najjar and P. K. Srimani, "A New Adaptive Hardware Tree-Based Multicast Routing in K-Ary N-Cubes," IEEE Trans. on Computers, vol. 50, no. 7, pp. 647–659, July 2001.
- [9] Z. Lu, B. Yi and A. Jantsch, "Connection-oriented Multicasting in Wormhole-switched Network-on-Chip," Proc. IEEE Comp. Society Annual Symposium on VLSI (ISVLSI'06), 6 pp., 2006.
- [10] J. Liu, L.-R. Zheng and H. Tenhunen, "Interconnect intellectual property for Network-on-Chip (NoC)," Journal of Systems Architecture, vol. 50, issue 2–3, Feb. 2004, pp. 65–79.
- [11] A. A. Chien and J. H. Kim, "Planar Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors," in Proc. of the 19th Int'l Symp. on Computer Architecture, pp. 268–277, May 1992.
- [12] F.A. Samman, T. Hollstein, and M. Glesner, "Multicast Parallel Pipeline Router Architecture for Network-on-Chip", in Proc. Design Automation and Test in Europe (DATE'08), pp. 1396–1401, 2008.
- [13] F. A. Samman, T. Hollstein and M. Glesner, "Networks-on-Chip based on Dynamic Wormhole Packet Identity Management". VLSI Design, Journal of Hindawi Pub. Corp., vol. 2009. (Article ID 941701), pp. 1–15, Jan 2009.
- [14] F.A. Samman, T. Hollstein, and M. Glesner, "Adaptive and Deadlock-Free Tree-Based Multicast Routing for Networks-on-Chip," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 18, no. 7, pp. 1067–1080, July 2010.
- [15] F.A. Samman, T. Hollstein, and M. Glesner, "New Theory for Deadlock-Free Multicast Routing in Wormhole-Switched Virtual-Channelless Networks-on-Chip". IEEE Trans. on Parallel and Distributed Systems, vol. 22, no. 4, pp. 544–557, April 2011.
- [16] F.A. Samman, T. Hollstein, and M. Glesner, "Wormhole Cut-Through Switching: Flit-Level Messages Interleaving for Virtual-Channelless Network-on-Chip". Elsevier Journal, Microprocessors and Microsystems – Embedded Hardware Design, vol. 35, no. 3, pp. 343–358, May 2011.
- [17] F. A. Samman, T. Hollstein and M. Glesner, "Planar adaptive network-on-chip supporting deadlock-free and efficient tree-based multicast routing method". Elsevier Journal, Microprocessors and Microsystems – Embedded Hardware Design, vol.36, no. 6, pp. 449–461, Aug. 2012.
- [18] F.A. Samman, T. Hollstein and M. Glesner, "Runtime Contention- and Bandwidth-Aware Adaptive Routing Selection Strategy for Networks-on-Chip". IEEE Trans. on Parallel and Distributed Systems, in press, available online, 22 June 2012 (doi:10.1109/TPDS.2012.200).

